Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 3 (2015) 1981 – 1988

Procedia MANUFACTURING

6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015

# Handling complexity of a model in system design: Framework, formalism and metrics

Hycham Aboutaleb*,Bruno Monsuez

*Department of Computer Science and System Engineering, ENSTA ParisTech828 boulevard les Maréchaux, Palaiseau 91120, France*

**Abstract**

Current systems complexity has reached a degree that requires addressing conception and design issues while taking into account all the necessary aspects. Therefore, one of the main challenges is the way complex system models are specified and designed. The exponential growing effort, cost, and time investment during the phase of modeling a complex systems emphasize the need for a paradigm, a framework, and an environment to handle the system model complexity. For that, it is necessary to understand the expectations of the human user of the model and his limits. This paper highlights the requirements a system model needs to fulfill to meet human user expectations.For that, it is necessary to be able to measure the system model complexity. This paper highlights the requirements a model needs to fulfill to match human user expectations, and suggests a graph-based formalism for modeling complex systems. Finally, a way to measure system model structural complexity based on Shannon theory of information is proposed.


Peer-review under responsibility of AHFE Conference

*Keywords:*Complexity metrics; Higraph; Hierarchy; System modeling

## 1. Introduction

Usually, the approach we follow in a project depends on how the results will be used. To optimize the design time, it is important to have a useful framework for analyzing complex systems and study their evolution. The use of such a framework requires an understanding of the boundaries of a given system, its components, its representation, and the evolution of its model and ways of representation.

* Corresponding author.
*E-mail address:*hycham.abou-taleb@ensta-paristech.fr


Peer-review under responsibility of AHFE Conference
doi:10.1016/j.promfg.2015.07.244

The complexity that emerges while designing and developing a system is usually the result of the multidimensionality of the system. To understand its behavior, a system is considered in the context of its environment, including interactions and interfaces. Indeed, the complexity of a system is often characterized, beyond the inherent complexity of components and their variety, by the complexity of the interaction network, from which emerges behaviors as intentional and unintentional that may be harmful and difficult to predict and control.

Since System Engineering is nowadays usually model-based, the more complex a system model is, the more difficult and expensive is the design and the implementation effort. However, little literature can be found about system model and architecture complexity. This is mainly due to the fact that large complex systems development projects are not repeatable, making comparative studies hard to perform. Moreover, there is no widely used system model complexity measure.

In this paper, we defined the requirements a system model needs to meet to be trustworthy, useful and understandable. The main addressed issue was the model complexity. First, this paper highlights the main issues in system modeling and a set of modeling requirements have been defined. Then it gives the mathematical definition of higraph model and introduces the underlying semantics.Another section summarizes the existing complexity measurements: an overview of the main complexity measurements is presented, including definitions and relevant properties. The next section defines metrics that are needed to evaluate a higraph-based model.

## 2. System representation and modeling

Model-based development has been adopted more or less in development of complex systems today. To understand this trend, it is necessary to focus on the properties of complex systems to design and to the needs of the stakeholders involved in the development of these complex systems. A model has to have a clear purpose: to help designing the system of interest. Modelers exclude all factors not relevant to the problem to ensure the project scope is feasible and the results timely. When the modeling process begins early on in the problem definition phase, the process can help the system designers focus their diagnosis on the system of interest.

### *2.1. Modelling Issues*

When developing complex systems, two main problems arise:

- The need to address all the aspects of the system of interest (to design and develop). [1]
- The need to share the knowledge between people involved in the process. [2]

To match these needs, model-based system engineering is helpful. However, the need of a model-based approach induces three new issues:

- Trustworthiness of the model: How close the model is to the reality?
- Understandability of the model: Is the model perceived and understood the same way by people?
- Usefulness of the model: Does the model help to get the desired results?

### *2.2. Trustworthiness of system model*

Given the limited cognitive capabilities of humans, we use models of the system properties and its context/environment that are of relevance and interest for the system design and development and disregard details considered irrelevant. A model is thus a deliberate simplification of reality with the objective of explaining a set of selected properties of the real system that are more important. This model starts first with a mental process to capture relevant information before the information captured is expressed through the means to be communicated. This information typically is the minimum information necessary to have a satisfactory understanding of the perceived real system and environment. [3]

### *2.3. Understandability of system model*

The understandability of a model depends on how an individual perceives the model that he/she is going to use. Two people share a similar mental model if they have similar descriptions, explanations, and predictions of the system of interest. Specifically, models allow people to similarly predict and explain the behavior of the system of interest, to recognize and remember relationships among its components and with its environment, and to construct expectations for what is likely to occur next.

### *2.4. Usefulness of system model*

A system model might be perceived differently since each one has his own mental map. Thus, this system model has a corresponding mental model in everybody's mind. To help ensure the utility of shared mental models, a distinction is often drawn among different types of mental models, normally based on their underlying content. In order to be useful, a mental model shall facilitate accomplishing a task and allow each system designer to work effectively as a member of the team [4]. Thus, a mental model would be considered effective if the team performance is increased. According to [5], a team performance is related to the taskwork mental model similarity, the teamwork mental model similarity, thetaskwork mental model perceived accuracy, and the teamwork mental model perceived accuracy. Moreover, it shall allow engineers to reuse and share past solutions. This has an additional advantage: inexperienced engineers benefit from the work of more experienced ones and are able to work at their quality levels.

## 3. Model complexity and hierarchy

### *3.1. Hierarchy issue*

Since systems are inherently complex, it is necessary to fully understand a system without reaching human mind limitations by handling this complexity. This system real complexity is indeed reflected in the corresponding system model. In fact, the system's perceived complexity is the model complexity. To obtain a model that is trustworthy, understandable, and useful, it is necessary to architect the complexity. As it is described in [6], there is a form of organized complexity in systems.

To handle large amounts of data, it is often useful to have a classification or an order. One effective way to classify a set of elements is to use a hierarchical organization of this set of elements, introducing sometimes a new order relation among the elements. With the hierarchy, in addition to be able to handle elements together, it becomes possible to handle subsets of elements together. There are two ways how to organize hierarchically a set: grouping and encapsulation.

- It is possible to group items based on similar properties or characteristics.
- It is possible to encapsulate many elements within a single element of a higher level and then consider only the properties of this element when an analysis is performed.

Therefore, to handle complexity of the real system, its model should be the result of a simplification strategy consisting of:

- Conceptual chunking: Refers to the formation of a higher-level concept that captures the essence of the problem-at-hand and reduces the complexity by omitting irrelevant details and reducing its dimensionality [7].
- Segmentation: Refers to the decomposition of a complex system into smaller parts that can be studied in isolation, in order that the capacity limitations of the human mind are avoided.

Consequently, we can identify two types of models hierarchies, generalization and aggregation:

- Generalization, i.e. hierarchy of types: The word type refers generally to a representation that gathers main properties of objects that have common characteristics [8]. One type allows to group elements with common characteristics. The mechanism of subtyping induces a hierarchy: an entity type T2, derived from type T1 has at least all the properties of an entity type T1.
- Aggregation: The word aggregation refers generally to a representation that gathers elements into another higher-level element to hide them when necessary. The higher-level element that encapsulates its elements has properties that are the emerging properties at this level due to the elements. Encapsulation also decreases the complexity of the system model [9]. Other names like "Nested Hierarchy" or "Container Hierarchy" are also common.

Finally, the hierarchy has an additional advantage: depending on the selected level, it is possible to observe different points of view.

### *3.2. Higraph-based model*

Graphs have been naturally used to represent and model problems since the emergence of computer science. To include both types of hierarchy identified above, higraph-based model is selected.

A higraph is a graph extended to include notions of depth and orthogonality and was introduced by Harel[10,11]. In other words:

*Higraph = Graph + Depth + Orthogonality*

#### *3.2.1. Definition(Higraph)*

A higraph is a quadruble $H = (B; E; \rho; \Pi)$ where :

- *B* is the set of blobs (or nodes);
- *E* is the set of edges.
- *ρ* is the hierarchy function. It assigns to each blob $b \in B$ its set of sub-blobs *ρ(b)*.
- *Π* is the orthogonality (or partitioning function) defined as $\Pi : B \to 2^{B \times B}$, associating with each blob $b \in B$ some equivalence relation *Π(b)* on the set of sub-blobs, *ρ(b)*.

By its definition, the depth, shown by a higraph is defined by the enclosure of one node within another.

## **4. Metrics**

### *4.1. Direct metrics*

Since complexity needs an unambiguous framework to be defined clearly and to be measured relevantly, basic metrics are identified first. In that purpose, a set of properties are identified as useful for the calculation of complexity of a higraph-based model.

#### *4.1.1. Size*

The most obvious and useful attributes of a model is its size, which can be measured statically for static as well as dynamic models. The most intuitive way is to take into account the number of nodes and the number of edges.

#### *4.1.2. Depth:*

The depth of a higraph-based model is the highest number of levels between the top node and the lowest level node.

#### *4.1.3. Width*

The width of a higraph-based model is the highest number of nodes at any one level.

### *4.2. Indirect metrics*

#### *4.2.1. Density:*

It measures the node constituents to the number of nested components. This metric is used to identify the density of nested elements.[12,13]

#### *4.2.2. Type Variety*

The number of types in a set of elements is a good indicator of variety if all the types are of equal importance, which is usually not the case[14].

This index is suitable since it possesses the following properties:

- For symmetric element types it equals the number of element types.
- The introduction or disappearance of a marginal type does not cause a discrete change in the variety index.

#### *4.2.3. Interface Load*

This index measures the average number of interface inputs into an element and the average number of interface outputs of an element and provides an overall measure for the whole model[15].

### *4.3. Shannon's entropy*

Statistical theory of information, as developed by Shannon[16], is an answer to the question: given a set of messages $\boldsymbol{m_i}$ each of which occurs with probability $\boldsymbol{p_i}$, what isthe amount of information they convey. The first step is to determine the amount of information provided by a single message $\boldsymbol{m_i}$, which is:

$$I(m_i) = -\log_2 p_i \tag{1}$$

#### *4.3.1. Definition (Shannon's entropy)*

Let then $X$ be a set of discrete random variables with values $x_1; x_2; ...; x_n$ with $x_i$ having probability $\mathrm{p_i}\ ; (1 < \mathrm{i} < \mathrm{n})$ Shannon's entropy $H$ is defined as:

$$H(X) = -\sum_1^n p_i \log_2 p_i \tag{2}$$

Consider a set $S$ containing $N_S$ states. We can split $S$ into $k$ independent subsets such that (Figure 1):

$$S = \bigcup_{i=1}^k S_i, S_i \neq 0, \forall i \tag{3}$$

And

$$N_S = \sum_{i=1}^k N_{S_i} \tag{4}$$

The probability of a state $x$ belonging to $S_i$ is:

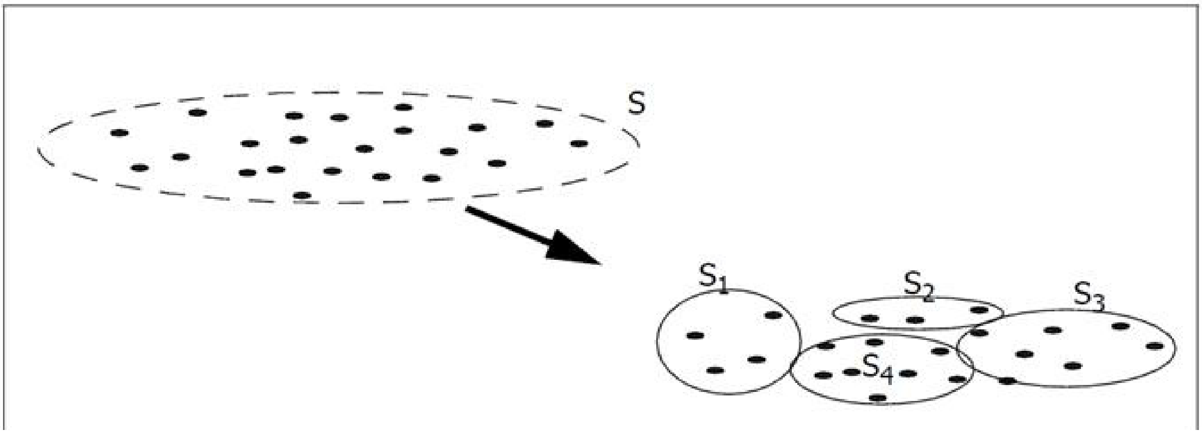


Fig.1.Decomposing a set.

$$p_i = -\sum_{i=1}^{k} p(x \in S_i) = N_{S_i} / N_S \tag{5}$$

The complexity of this system is thus:

$$H(S) = -\sum_{i=1}^{k} p_i \log_2 p_i = -\sum_{i=1}^{k} N_{S_i} / N_S \log_2 (N_{S_i} / N_S) \tag{6}$$

By changing the perspective from working with a large set $S$ of $N_S$ individual states ($x$) to a collection of subsets containing a smaller number $N_{S_i}$ of states ($x \in S_i$), the whole set complexity has been replaced with the probability weighted sum of the complexity found within each subset. This is a very powerful principle in design: a complex problem is decomposed into a set of smaller problems with smaller complexity. Besides, the global complexity is the same.

## 5. Higraph-based system model structural complexity

A higraph model $M$ entropy intuitively depends on the number of blobs, the number of edges, the hierarchy and the orthogonality [17].

We use Shannon's entropy as an indicator of the complexity.

We get the entropy of the model higraph as follows:

$$H = H_B + H_E + H_\rho + H_\Pi \tag{7}$$

To evaluate the complexity of a higraph $M$ , it is consequently necessary to get the complexity get each term separately.

- $H_B$ :

$$H_B = H(B) = -\log_2(1/|B|) = \log_2(|B|) \tag{8}$$

- $H_E$ :

$$H_E = H(E) = -2\log_2(1/|E|) = 2\log_2(|E|) \tag{9}$$

It takes into account the head and the tail of the edge.

- $H_\rho$ :

$H_\rho$ relates to the number of hierarchical relationships between the elements of the model $N$ .
Multiple locations of an element, i.e. an element has several parents, are taken into account.

$$N = \sum_{x \in M} |\rho(x)| \tag{10}$$

It is obvious that if there is no hierarchy, $N = |B|$, i.e. the diagram contains all the elements at the same level.

$$H_\rho = -2\log_2(1/|N|) = 2\log_2(|N|) = 2\log_2(\sum_{x \in M} |\rho(x)|) \tag{11}$$

Where we take into account parent and child relationship.

- $H_\Pi$ :

$H_\Pi = H(M_\Pi)$ , where $M_\Pi$ is the Type Higraph associated to the higraph $M$ .
Let $M_\Pi$ be a Type Matrix higraph.
Let $M$ be a Model higraph.
Let $g : M \to M_\Pi$ a morphism that associates to each element (object, flow, attribute) *x* of the Model higraph $M$ to its type, with $M_\Pi$ , the Model Type Higraph.
We have:

- $\forall x \in M, g(x) \in M_\Pi$;
- $\forall x \in M, g(\rho(x)) \subset \rho(g(x))$;
- $\forall t \in M_\Pi, g(\Pi_t(x) \subset \rho(t)$.

Besides, $H_{\Pi} = (B_{M_{\Pi}}; E_{M_{\Pi}}; \rho; \Pi)$ have the following properties:

- $E_{M_{\Pi}} = 0$, i.e. there is no edge;
- $\forall x \in B_{M_{\Pi}}, \Pi(x) = \rho(x)$; i.e. all elements are of the same type.

We have:

$$H_{\Pi} = H(B_{M_{\Pi}}) + H(E_{M_{\Pi}}) + H_{\rho}(M_{\Pi}) + H_{\Pi}(M_{\Pi}),$$

where:

- $H(B_{M_{\Pi}}) = \log_2 \left|B_{M_{\Pi}}\right|$
- $H(E_{M_{\Pi}}) = 0$
- $H_{\rho}(M_{\Pi}) = 2\log_2(\sum_{x \in M_{\Pi}} \left|\rho(x)\right|)$
- $H_{\Pi}(M_{\Pi}) = 0$

Thus, we get the entropy of the model higraph as follows:

$$H = H_B + H_E + H_{\rho} + H_{\Pi}$$

i.e.

$$H = \log_2 |B| + 2\log_2 |E| + 2\log_2 (\sum_{x \in B} |\rho(x)|) + \log_2 |B_{\Pi}| + 2\log_2 (\sum_{x \in M_{\Pi}} |\rho(x)|) \quad (12)$$

## References


[1] Jorgensen, H.D. “Interactive Process Models.” PhD thesis, Norwegian University of Science and Technology Trondheim, Norway, 2004

[2] Avnet, M.S. “Socio-Cognitive Analysis of Engineering Systems Design: Shared Knowledge, Process, and Product.” PhD thesis, MIT, USA, 2009.

[3] Mostashari, A. “Stakeholder-Assisted Modeling and Policy Design Process for Engineering Systems.” PhD thesis, MIT, USA, 2005.

[4] Cannon-Bowers, J.A.; Salas, E.; Converse, S.A. 1993. “Shared mental models in team decision making.” Individual and Group Decision Making, pages 221–246. N.J. Castellan Jr.

[5] Forsberg, K.; Harold Mooz, H. 2000. “The relationship of systems engineering to the project cycle.” Journal of Applied Psychology, 85(2):273–283.

[6] Simon H.A. 1962. “The architecture of complexity.” Proceedings of the American Philosophical Society, 106(6):467–482.

[7] Halford, G.S; Baker, R; McCredden, J.E; Bain, J.D. 2005. “How many variables can humans process?” Psychological Science, 16(1):70–76.

[8] Cardelli, L; Wegner, P. 1985. “On understanding types, data abstraction,and polymorphism.” ACM Comput.Surv., 17(4):471–522.

[9] Ahl,V; Allen, T. F. H. 1996. “Hierarchy Theory - A Vision, Vocabulary, and Epistemology.”Columbia University Press.

[10] Harel D. Statecharts: A visual formalism for complex systems. Science of Computer Programming, 8(5):231–274, 1987.

[11] Harel D. On visual formalisms. Communications of the ACM, 31(5):514–530, 1988

[12] Kumar S, Tomar P, Nagar R, Yadav S. Coupling metric to measure the complexity of component based software through interfaces. International Journal of Advanced Research in Computer Science and Software Engineering, 4(4):157162, 2014.

[13] Narasimhan VL, Hendradjaya B. Some theoretical considerations for a suite of metrics for the integration of software components. Journal Information Sciences, 177(3):844864, 2007.

[14] Straathof. SM Shannon's entropy as an index of product variety. Economic Letters, 94(2):297303, 2007.

[15] Kumari U, Upadhyaya S. An interface complexity measure for component-based software systems. International Journal of Computer Applications, 36(1):4652, 2011.

[16] Shannon CE. A mathematical theory of communication. Bell SystemTechnical Journal, 64:379–423,623–656, 1948.

[17] Rossi M, Brinkkemper S. Complexity metrics for systems development methods and techniques. Information Systems, 21(2):209–227,1996